

EmployeeLock

A hash-chained accountability workbook

Paper ID: EL-WP-0.1 · Product: EmployeeLock v0

Aziel / GodLock.AZ · 2 September 2026 · License: Apache-2.0

Name the event. Name who owns the record. Keep the leftover blame blank if it is blank. Chain the row.

Abstract

EmployeeLock is local software plus a spreadsheet. It logs operational events so they cannot sit unnamed. Each row carries five working fields on one line — event, result, blame placed, short-term outcome, long-term outcome — plus a separate owner of the record. File imports (audio, video, photo, document, other) are hashed and indexed. Rows are linked with SHA-256 in the TemporalLock ethic.

Unowned Lattice (UL / BAL) is the issue cluster this tool is built to fight: lattice / unowned / renamed. UL is not a Lock module. This paper does not file EmployeeLock under UL-CAT and does not replace BAL-WP, BAL-IS, BAL-WN, or BAL-TR. It is a Lock module that makes unowned and renamed rows visible.

Runtime and workbook shipped with this paper: `employeelock.py` and `EmployeeLock_v0.xlsx`. No public repository or DOI is claimed as of this paper.

1. Purpose

Events happen. Files exist. Outcomes fade. Names move. The record later says “the process” did it.

EmployeeLock exists so that a row has to answer six questions and then survive a hash check:

- What happened. (event)
- What followed. (result)
- Who or what was named as responsible. (blame_placed — blank is allowed and is itself recorded)
- What happened soon. (outcome_short)
- What lasted. (outcome_long)
- Who owns this record. (owner_named — blank flags UNOWNED)

Blame and ownership are not the same field. A row can name a blame target and still be unowned as a record. A row can be owned and still leave blame blank.

The design goals:

- One line, five working fields. Do not split the story across hidden tabs and call it complete.
- Unowned is visible. Do not let a blank owner look like a completed record.
- Renamed is visible. renamed_from is first-class.
- Media is hashed, not embedded. The workbook is an index.
- Append, don't edit hashed cells. A later long outcome is a new row that cites the old row_hash.
- Verify bytes, not narratives. verify recomputes hashes. It does not decide who was right.

2. What this is / is not

This is

- a workbook (COVER, LOG, EVIDENCE, CHAIN, OWNERS, DASH, LISTS);
- a CLI that inits, appends, imports, and verifies;
- a linear hash chain of rows;
- a countermeasure against the UL surface (unowned / renamed), not a member of that catalog.

This is not

- UL, and not a BAL issue paper;
- FoldLock (compression);
- TemporalLock (this borrows the ethic and stays a different product);
- a court filing, exhibit stickerer, or counsel;
- a truth score, consensus layer, or token;
- a remote uploader or anonymous relay;
- a charge sheet against a named living person in the shipped demo.

The shipped workbook’s two demo rows are generic (“records desk”, “unnamed prior process”). They are format proof, not case facts.

3. How it fights the UL surface

UL tracks three beats. EmployeeLock answers each beat with a field, not with a speech.

UL beat	EmployeeLock answer
Lattice — work spread across unnamed structure	One LOG row holds event × result × blame × short × long.
Unowned — no one owns the act or the record	owner_named blank → formula flag UNOWNED. DASH counts those rows.
Renamed — the name that would have held still is moved	renamed_from filled → formula flag RENAMED.

Citation of BAL papers stays by BAL ID when the issue itself is discussed. This product is EL-WP-0.1.

4. Workbook sheets

Spec string: employeelock-v0.

4.1 COVER

Operator instructions. Spec line. CLI examples. Boundary statements. No legal name. No home location.

4.2 LOG

Append-only event register. Headers at row 4. Data from row 5. Capacity in v0: 500 prefilled formula rows (MAX_LOG). Freeze at C5. Autofilter on the header row. Landscape / tabloid print setup. Input cells use blue font. Formula flags do not.

4.3 EVIDENCE

File index. One row per imported file. Does not embed audio, video, or photos. Stores path + SHA-256 + size + kind + linked entry_id.

4.4 CHAIN

Mirror of LOG link fields for reading the chain without the working columns. link_ok pulled from LOG.

4.5 OWNERS

Slots for distinct owner_named values the CLI has seen. Counts rows owned and times the same string appears in blame_placed. Separate cells count UNOWNED and RENAMED over the data range.

4.6 DASH

Counts: events, evidence files, owned, unowned, renamed, chain OK, genesis, breaks, blank blame, blank long outcome. Bar chart of those counts.

4.7 LISTS

kind: audio, video, photo, document, other, none. exhibit_potential: Y, N, maybe. observer_hint: operator, witness, system, third-party.

5. LOG columns

Order is the v0 contract. Do not reorder without a version bump.

Col	Name	Source	Role
A	entry_id	CLI	EL-0001...
B	timestamp_utc	CLI	UTC YYYY-MM-DDTHH:MM:SSZ
C	event	operator	Required to count as a row
D	result	operator	What followed
E	blame_placed	operator	Responsible name or blank
F	owner_named	operator	Owner of the record
G	unowned	formula	UNOWNED / owned / blank
H	renamed_from	operator	Prior name, if any
I	name_moved	formula	RENAMED / stable / blank
J	outcome_short	operator	Near effect
K	outcome_long	operator	Lasting effect
L	cites	operator	Prior row_hash when this row extends an older one
M	confidence	operator	Observer-assigned 0.0–1.0
N	observer	operator	Validation list
O	exhibit_potential	operator	Y / N / maybe. Not an exhibit number.
P	evidence_ids	CLI	Comma-joined EV-...
Q	file_sha256s	CLI	Comma-joined file digests
R	canonical	CLI	Canonical JSON string that was hashed

Col	Name	Source	Role
S	prev_hash	CLI	Prior row_hash or 64 zeros
T	row_hash	CLI	SHA-256 hex of canonical
U	link_ok	formula	GENESIS / OK / BREAK

Formulas (v0)

```
G: =IF(C{r}="", "", IF(TRIM(F{r})="", "UNOWNED", "owned"))
I: =IF(C{r}="", "", IF(TRIM(H{r})="", "stable", "RENAMED"))
U5: =IF(C5="", "", IF(S5="<64 zeros>", "GENESIS", "BROKEN"))
U{r>5}: =IF(C{r}="", "", IF(S{r}=T{r-1}, "OK", "BREAK"))
```

DASH / OWNERS counts use G5:G504, I5:I504, U5:U504 so the header word “unowned” is not counted. Excel COUNTIF is case-insensitive. Conditional color: UNOWNED and BREAK red, owned and OK green, RENAMED amber, GENESIS gold-tan.

6. EVIDENCE columns

Col	Name	Rule
A	evidence_id	EV-0001...
B	imported_utc	Same timestamp as the LOG row that received the file
C	kind	List, or guessed from suffix on import
D	original_name	Filename only
E	stored_path	Absolute path the operator still controls
F	size_bytes	Integer
G	file_sha256	Hex digest of file bytes
H	linked_entry_id	The LOG entry_id
I	note	Optional

Kind guess from suffix: audio .wav .mp3 .m4a .flac .ogg .aac; video .mp4 .mov .mkv .webm .avi; photo .jpg .jpeg .png .gif .webp .tif .tiff .heic; document .pdf .doc .docx .txt .md .xlsx .csv; else other. Override with --kind.

The workbook does not copy the file into itself. v0 has no --copy-to. Operator keeps the bytes. verify re-hashes the path if the file is still there and reports missing_files if it is not. A missing file is a location problem, not automatically a chain break.

7. Canonical encoding and row hash

Hash function: SHA-256, lowercase hex. Canonical payload is UTF-8 JSON with sorted keys and compact separators. confidence is forced to exactly six decimal places.

Hashed fields, and only these:

```
entry_id, timestamp, event, result, blame_placed,
owner_named, renamed_from, outcome_short, outcome_long,
evidence_ids, confidence, observer, file_sha256s, prev_hash
```

Not hashed: unowned, name_moved, link_ok, cites, exhibit_potential, canonical, row_hash. Those are either derived or operator-side pointers. If cites or exhibit_potential need to enter the digest later, bump the spec string.

Genesis prev_hash is 64 ASCII zero characters. The next row's prev_hash must equal this row's row_hash. A second workbook that starts from the same genesis and diverges is a fork. v0 does not merge forks. Each file is one linear chain.

8. CLI

```
python3 employeelock.py init    WORKBOOK.xlsx
python3 employeelock.py append WORKBOOK.xlsx \
    --event TEXT --result TEXT --blame TEXT --owner TEXT \
    --short TEXT --long TEXT [--renamed-from TEXT] [--cites HASH] \
    [--confidence 0.7] [--observer operator] [--exhibit Y|N|maybe]
python3 employeelock.py import WORKBOOK.xlsx FILE [FILE ...] \
    --event TEXT --result TEXT --blame TEXT --owner TEXT \
    --short TEXT --long TEXT [--kind KIND] [--note TEXT] ...
python3 employeelock.py verify WORKBOOK.xlsx
```

init builds a fresh workbook. append writes a LOG row with no files. import writes one LOG row plus one EVIDENCE row per file, all sharing that entry_id. verify walks LOG, recomputes each row_hash, checks prev_hash linkage, and re-hashes evidence files that still exist at stored_path.

Exit code 0 if the chain hashes. Nonzero if any row hash or link fails. Missing files are reported and do not by themselves fail the row-hash check. Do not type hashed cells by hand and expect verify to pass.

9. Verification result

```
{
  "ok": true|false,
  "rows": int,
  "errors": [...],
  "missing_files": [...],
  "unowned": int
}
```

ok is false when a stored row_hash does not match the recomputed digest or when prev_hash does not match the previous stored digest (or genesis zeros on the first row). unowned counts rows whose owner_named is blank. It is a surface metric, not an error.

10. Path contract

v0 hashed cells are Path B: the CLI appends. It does not rewrite a prior row's hashed fields. If a long outcome arrives later, append a new row. Put the earlier row_hash in cites. Leave the old row readable. Manual edits to event text after a write break verify. That is the point.

Path A (rebuild the sheet from a seed of events) is not shipped. A later version may add an export/rebuild. It must mint new entry_id values or a new workbook, not silently retarget old hashes.

11. Worked miniature (informative)

Shipped demo after init + one append + one import of this product's own spec file (hashes change on every rebuild because timestamps change; structure does not):

- EL-0001 genesis. Owner blank → UNOWNED. link_ok = GENESIS. Event: process outcome recorded with no named owner.

- EL-0002 owned by “records desk”. renamed_from = “ticket queue” → RENAMED. Evidence EV-0001 = EmployeeLock_v0_spec.md, kind document, file SHA-256 bound to the row. link_ok = OK.
- DASH after recalc: 2 events, 1 evidence file, 1 owned, 1 unowned, 1 renamed, 1 genesis, 1 OK, 0 breaks.

Those two rows travel with the workbook so the sheet is not empty. Replace them with real work. Do not treat them as accusations.

12. Relation to the rest of the mesh

Sibling	Boundary
FoldLock	Compress / decompress. EmployeeLock may index a .fld as a file. It does not fold.
TemporalLock	Time receipts. EmployeeLock is a row chain about events and owners, not a general observation log.
ForgeReceipts	Packaging. A verified workbook can be a payload.
EmbryoLock / ARK1	Wrap a copy if the threat model says so. Do not phoenix-overwrite a Path B original that must stay byte-stable.
GodLock	Public ABAD node. Not this sheet.
UL / BAL	Issue papers stay issue papers. EmployeeLock cites the cluster and refuses to be filed as UL.

13. Limits

EmployeeLock v0 does not:

- embed media binaries in the xlsx;
- assign exhibit numbers;
- score blame or compute a “truth” percentage from confidence;
- merge two workbooks;
- talk to a network;
- store operator passwords;
- place a legal name or residential location in the template;
- speak Horton v. Horton facts (that work has its own skill);
- treat a blank blame_placed as a defect — only a blank owner_named is flagged unowned.

Capacity is 500 LOG rows and 500 EVIDENCE rows in the prefilled formula window. Beyond that, bump the builder.

14. Status

Item	State
Structure name	EmployeeLock v0
Paper	EL-WP-0.1 — this document
Companion spec	EmployeeLock_v0_spec.md
Workbook	EmployeeLock_v0.xlsx
Runtime	employeelock.py

Item	State
Spec string	employeelock-v0
Public repo	None claimed
DOI	None claimed
License	Apache-2.0

Aziel / GodLock.AZ
2 September 2026

A fork that hides unowned rows, drops file hashes, elects a blame score, or files this under UL-CAT is no longer this spec.